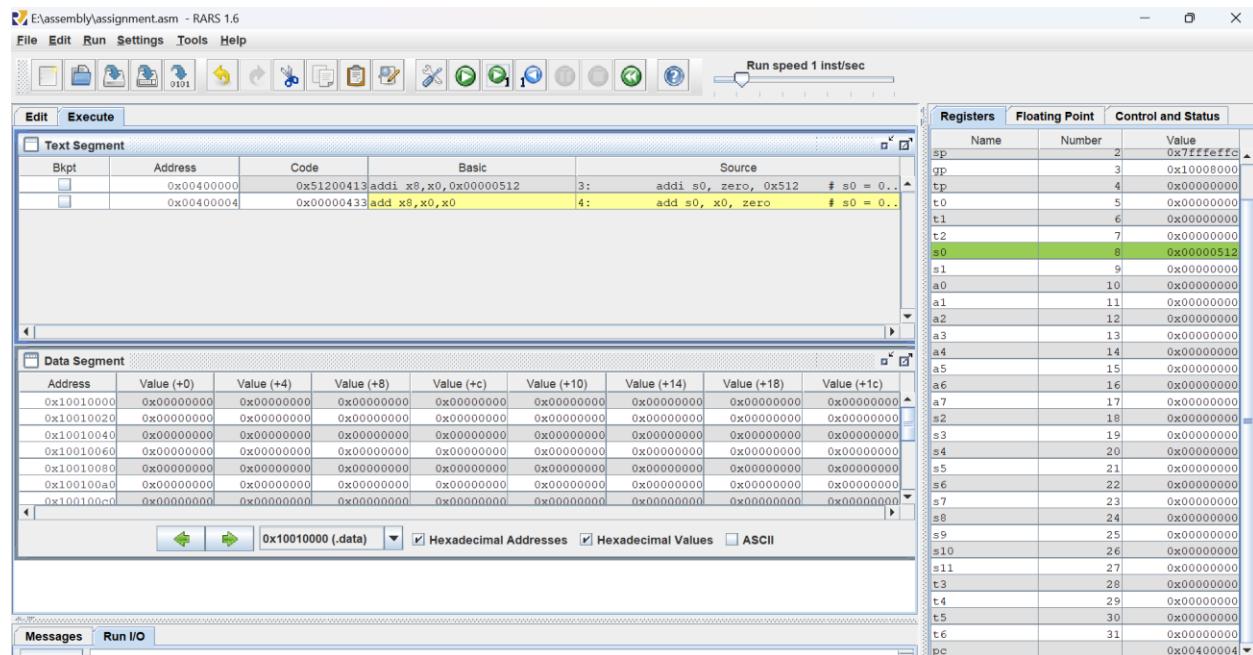
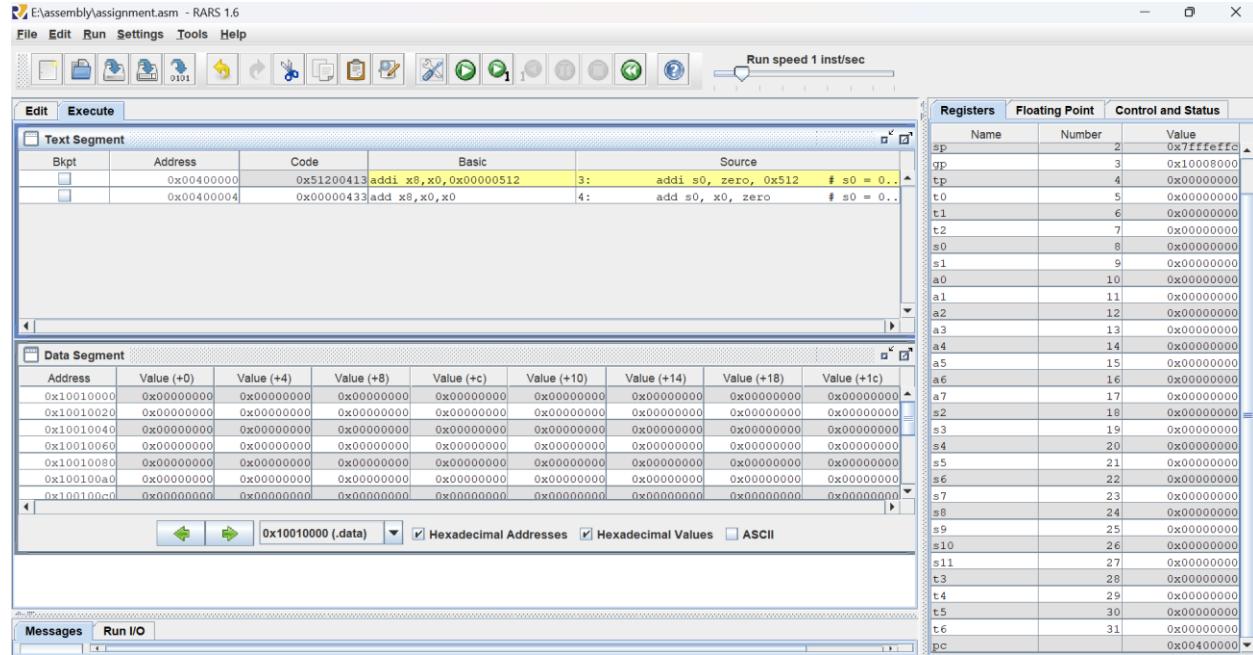


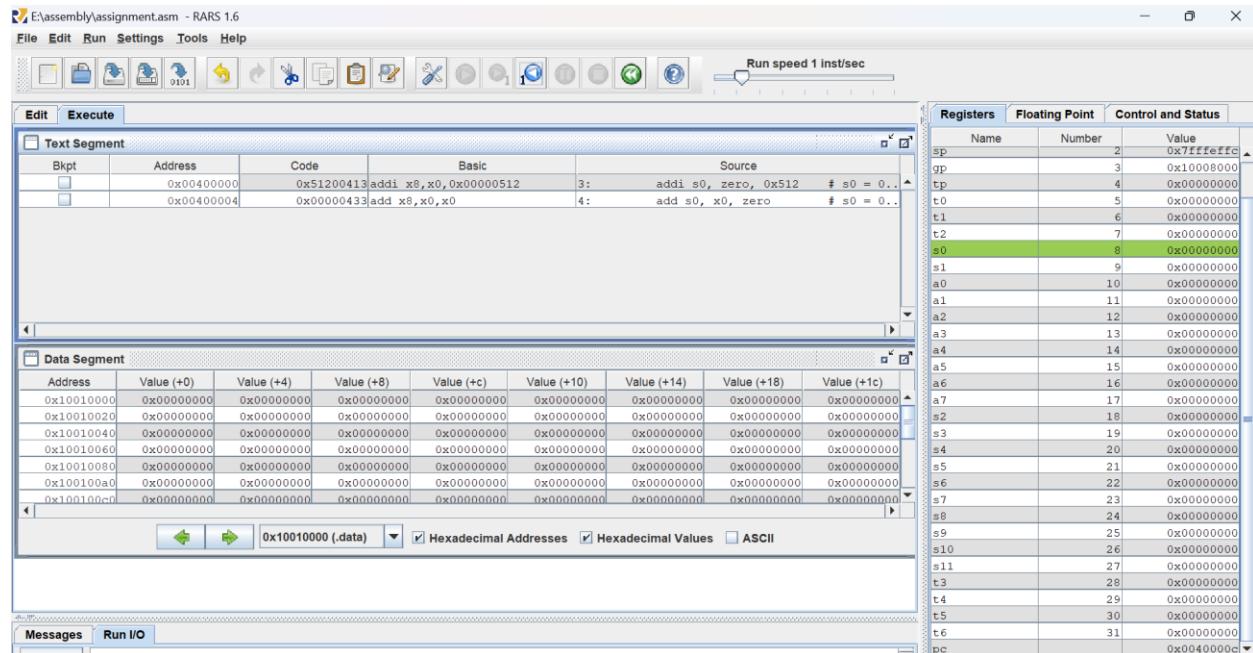
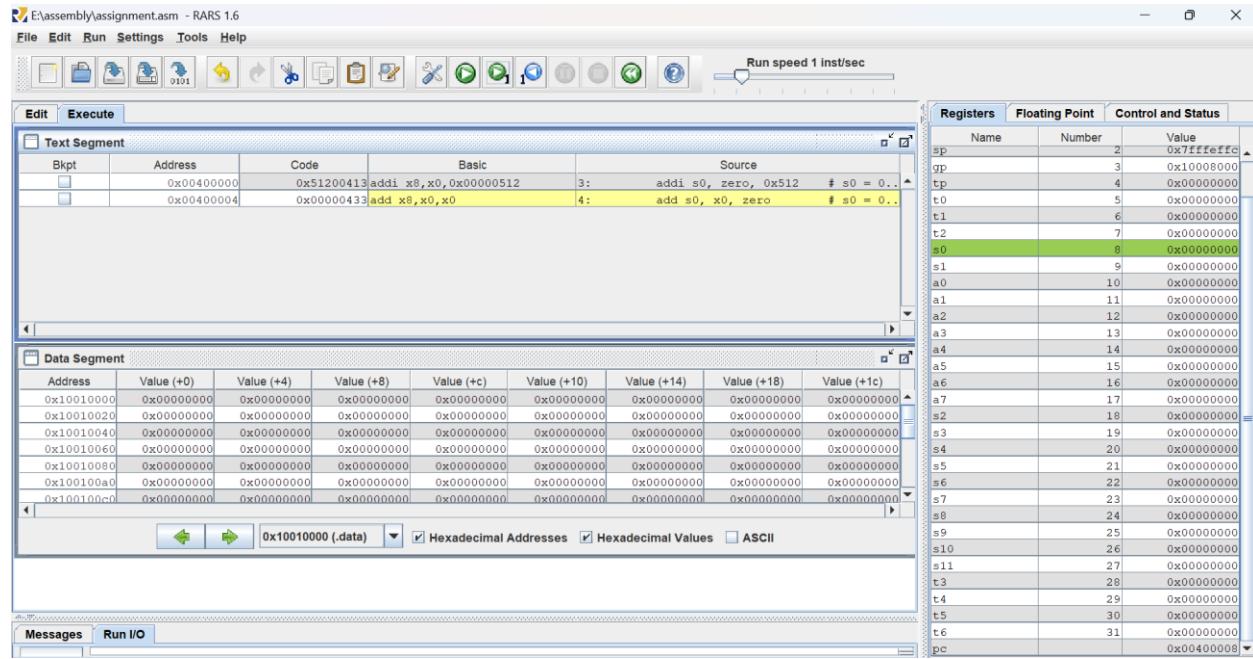
Student name: Lê Ngọc Anh Vũ

Student ID: 20236014

Assignment 1: Assign 12-bit integer numbers / small integer

- Run step by step and monitor the Registers window:





- Learning more about instructions **lb, sb**:

lb (Load Byte) instruction is used to load a byte from memory into a register.

sb (Store Byte) instruction is used to store a byte from a register into memory.

- In Text Segment, the instruction formats of the machine codes are: 0x51200413, 0x00000433.
- If we replace 0x512 by 0x20232024, there is nothing happening:

The screenshot shows the RARS 1.6 debugger interface. The assembly code window contains the following instructions:

```

1 # Laboratory Exercise 2, Assignment 1
2 .text
3 addi s0, zero, 0x20232024    # s0 = 0 + 0x20232024; I-type: just store a constant with 12-bit length
4 add s0, x0, zero      # s0 = 0 + 0; R-type:
5

```

The Registers window on the right shows the state of various registers:

Name	Number	Value
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x04000000

This is because addi instruction is just used to assign a small integer in the range of 12-bits (from -2048 to 2047) but 0x20232024 is out of range of 12-bits. As a result, nothing happens.

Assignment 2: Assign 32-bit integer

- Run step by step and monitor the Registers window:

The screenshot shows the RARS 1.6 debugger interface during execution. The assembly code window shows the following instructions:

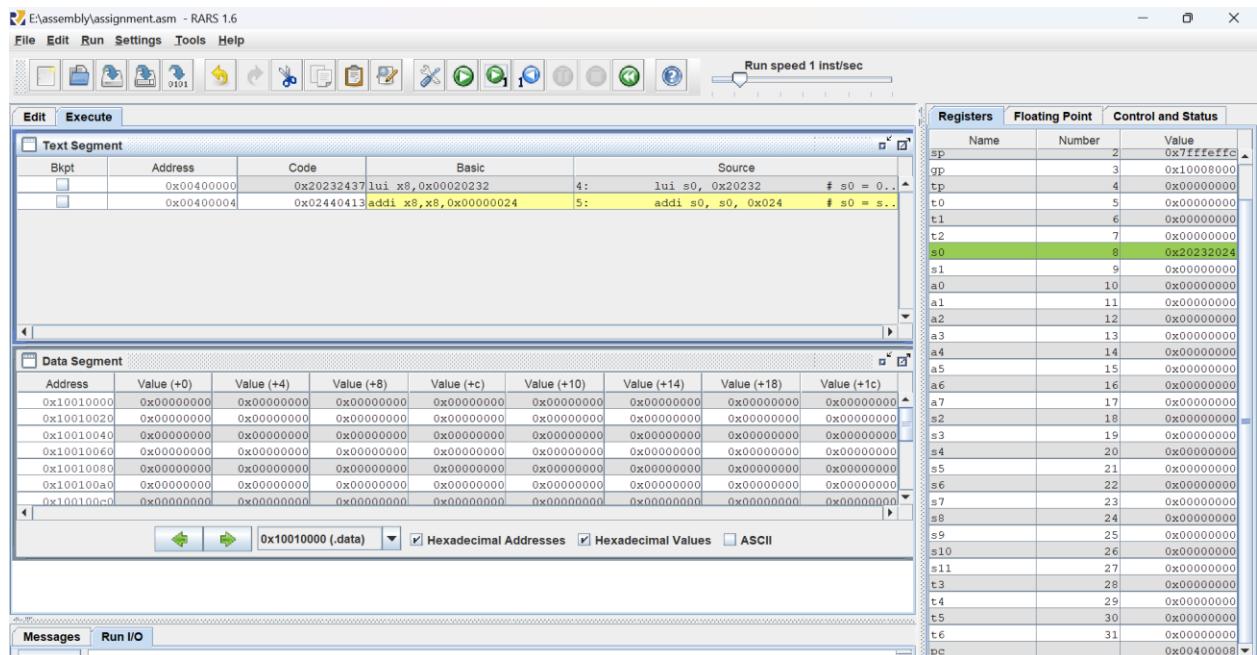
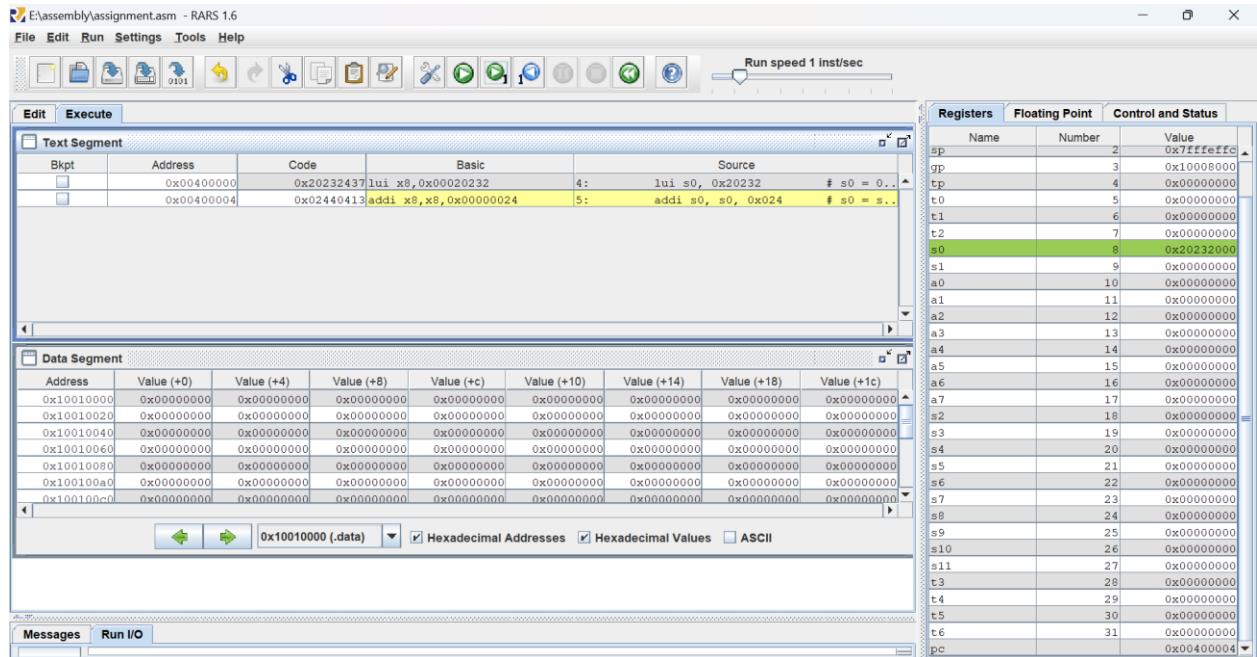
```

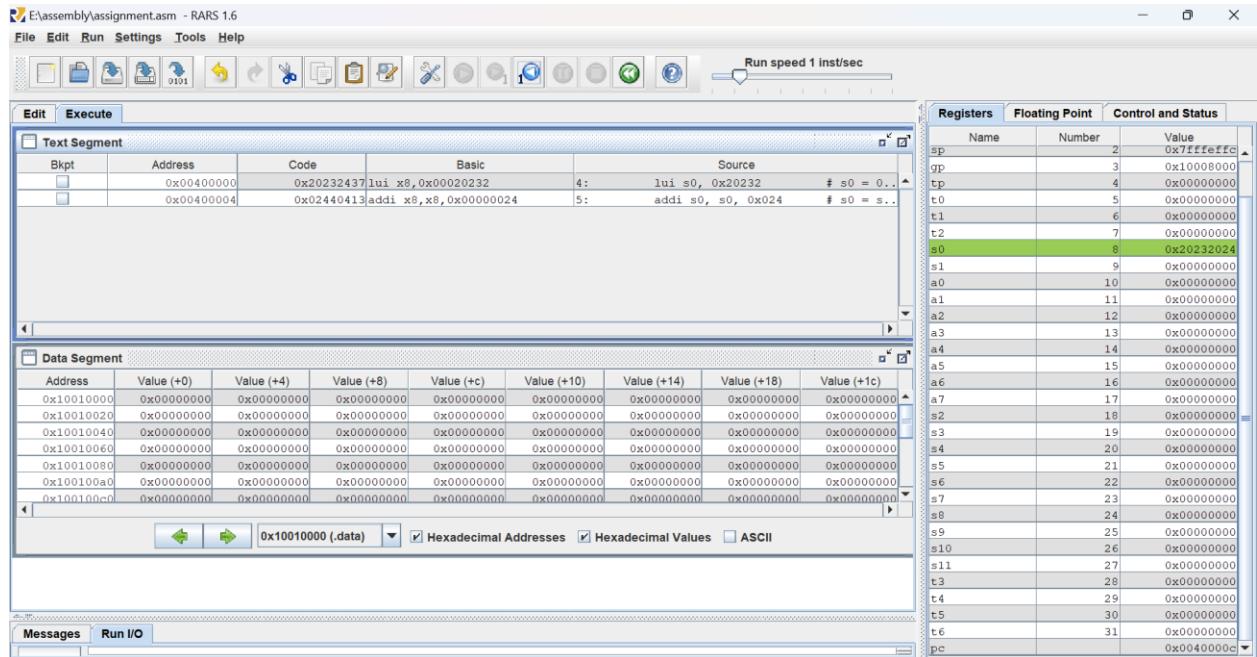
Text Segment
Bkpt Address Code Basic Source
4: 0x04000000 0x20232437 lui x8,0x00020232
5: 0x00400004 0x02440413 addi s0, x8,x8,0x00000024

```

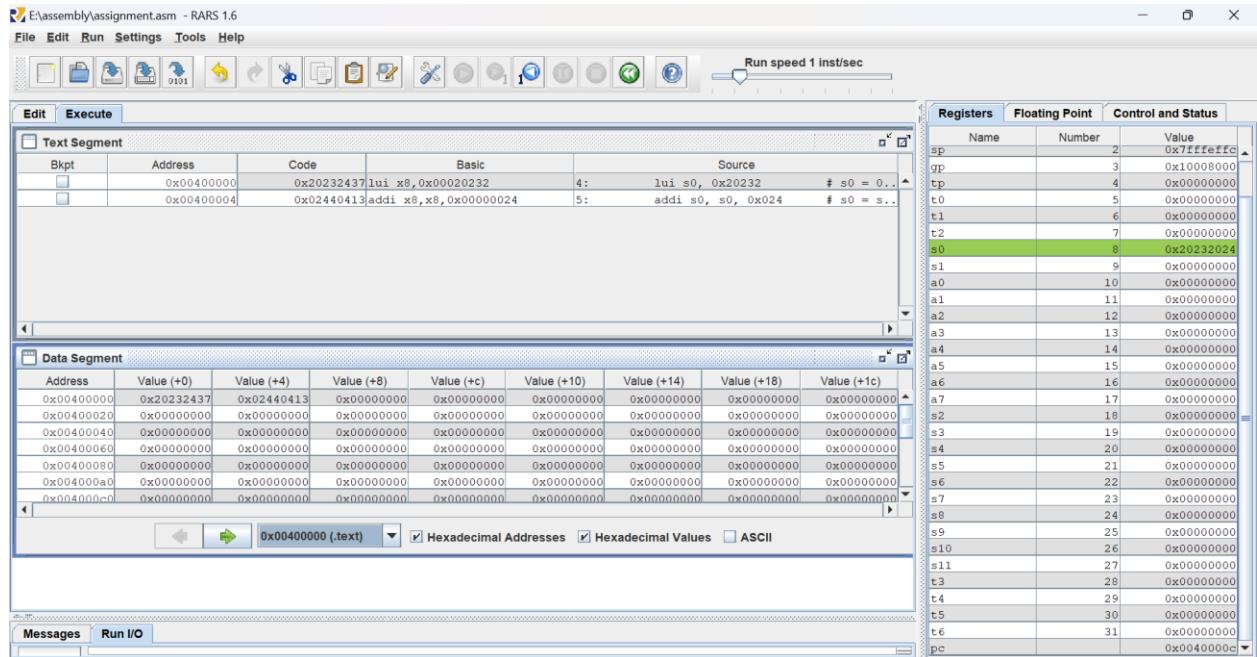
The Registers window on the right shows the state of various registers:

Name	Number	Value
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x04000000



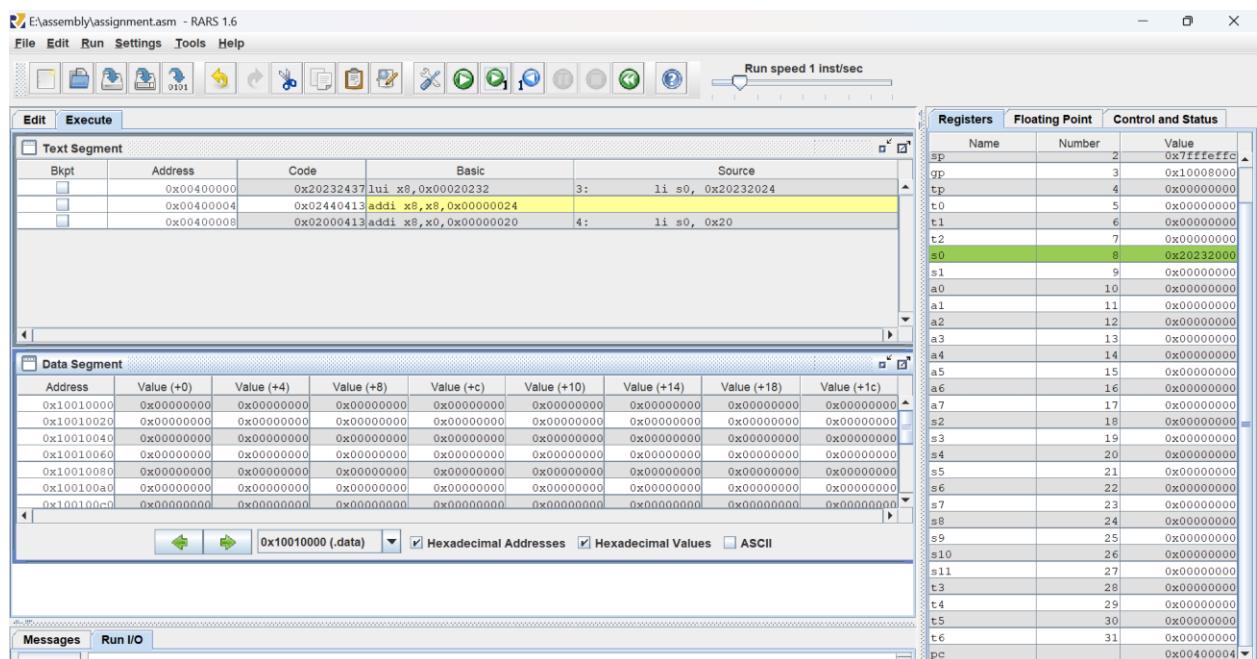
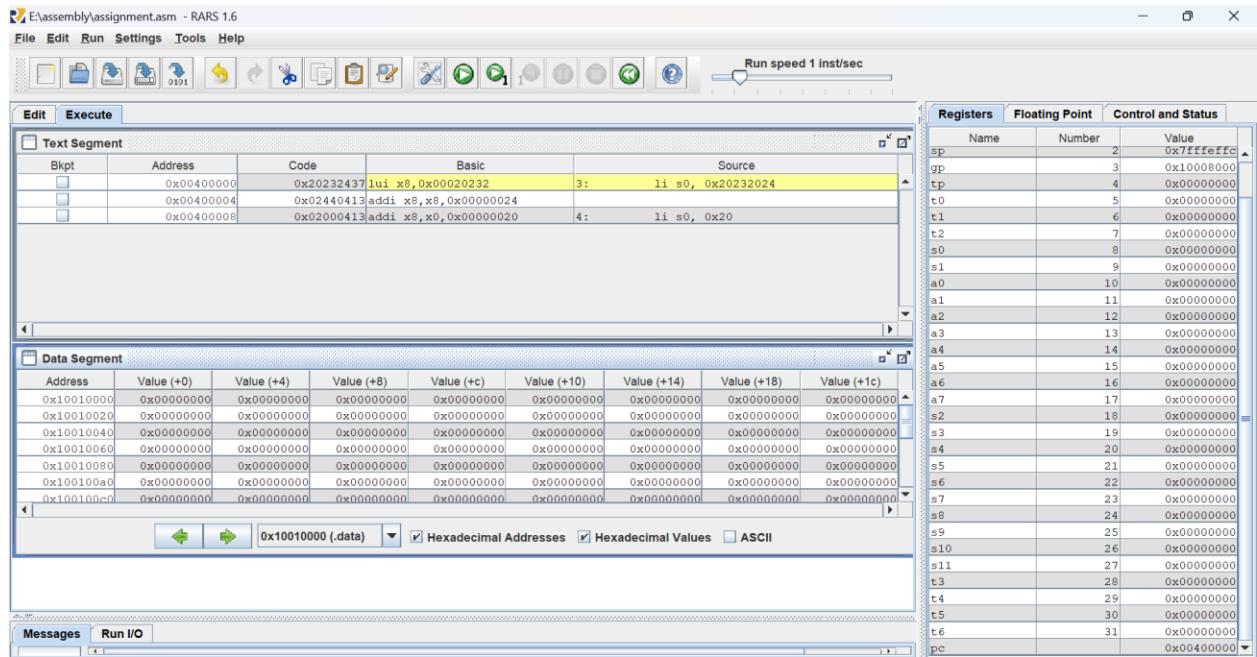


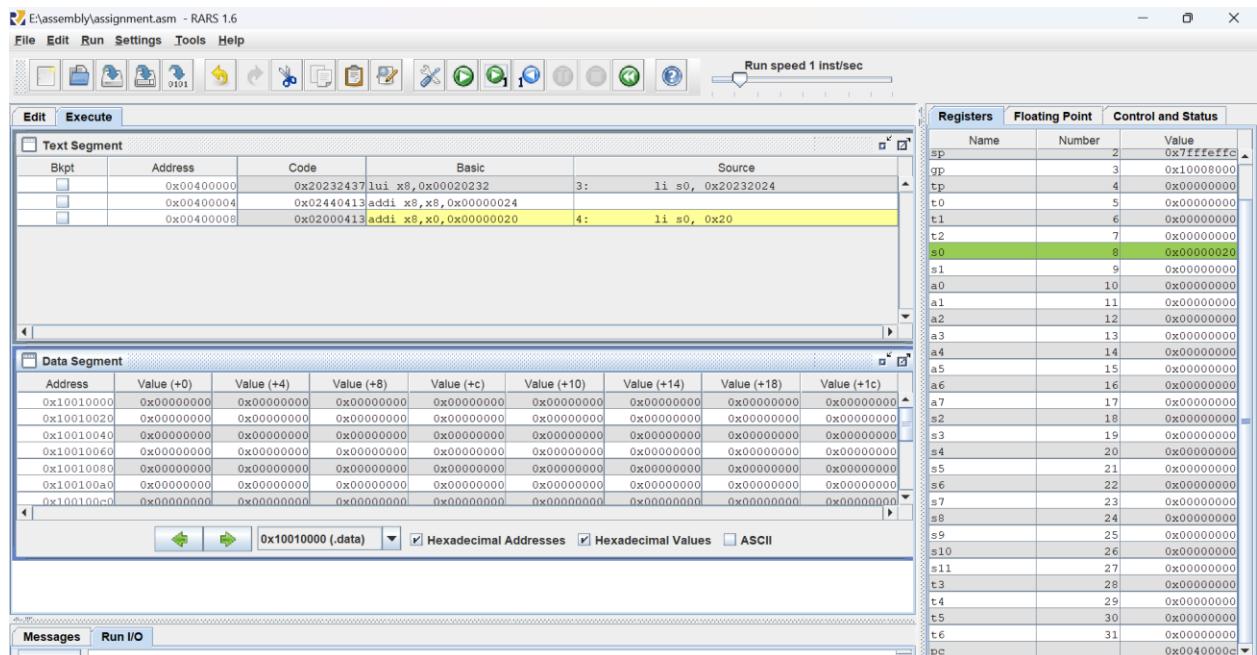
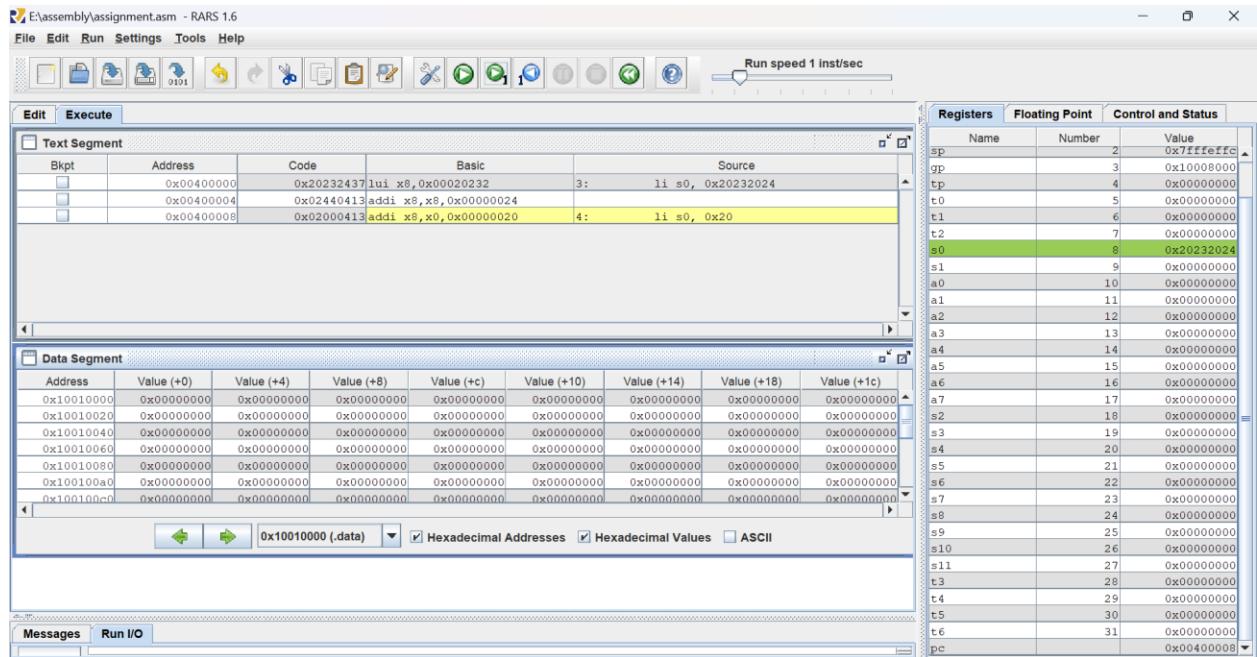
- In Text Segment, the instruction formats of the machine codes are: 0x20232437, 0x02440413.
- Data in Data Segment at first two cells is the same with machine codes in Text Segment:

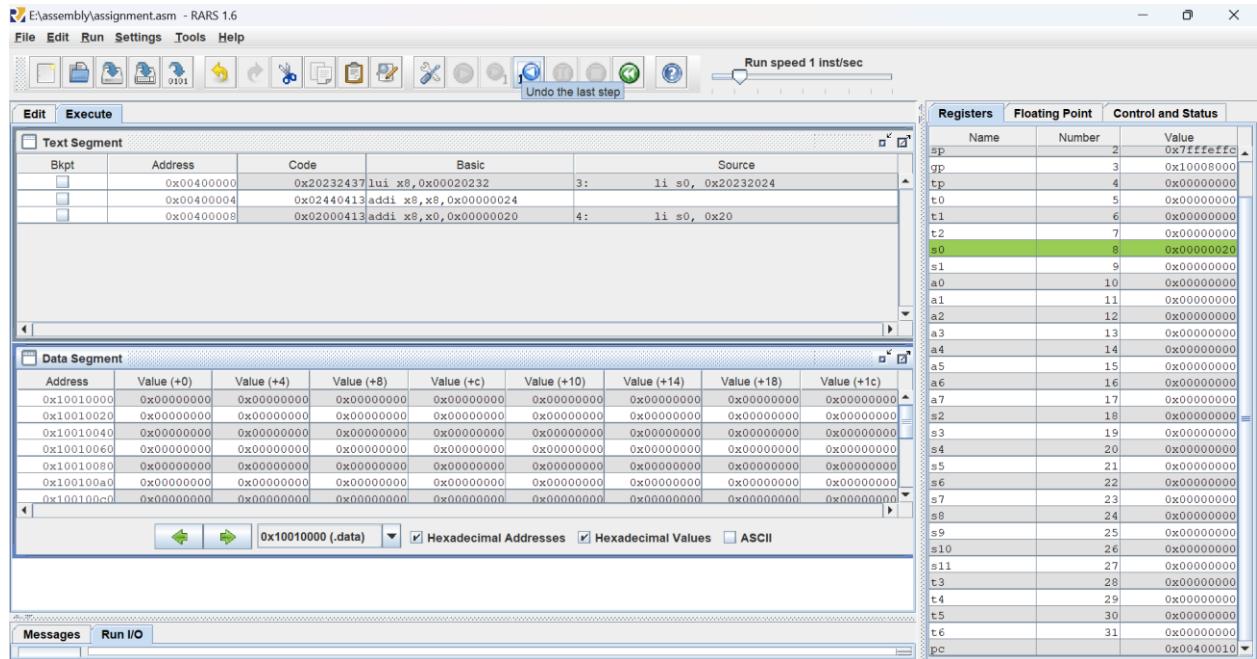


Assignment 3: new assignment instructions

- Compile and observe step by step:



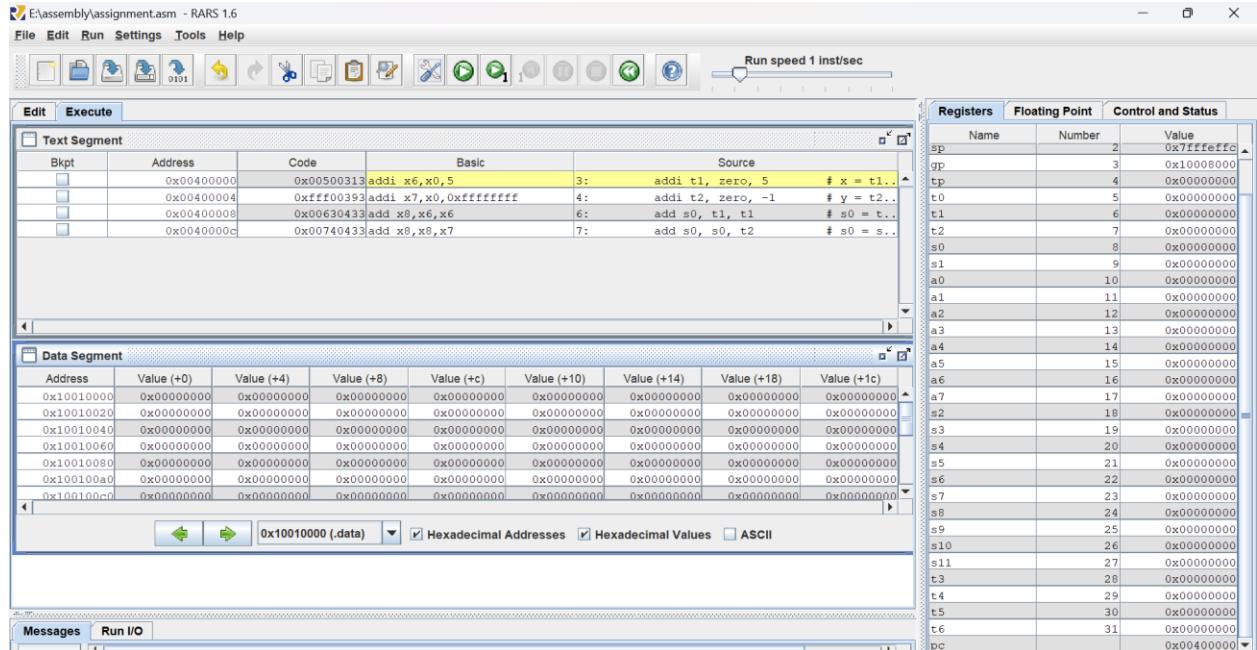




In Text Segment window, there are 3 commands in Basic column and 2 ones in Source column. This is because Basic column contains a extended instruction “li s0, 0x20232024 which includes two other ones. They are “lui x8, 0x00020232” and “addi x8, x8, 0x00000024”. Consequently, Basic column has more 1 command than Source column.

Assignment 4: Calculate the expression $2x + y = ?$

- Run step by step and monitor the Registers window:



E:\Assembly\assignment.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed 1 Inst/sec

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00500313 addi x6,x0,5	3:	addi t1, zero, 5 # x = t1..
	0x00400004	0xffff00393 addi x7,x0,0xffffffff	4:	addi t2, zero, -1 # y = t2..
	0x00400008	0x00630433 add x8,x6,x6	6:	add s0, t1, t1 # s0 = t..
	0x004000c	0x00740433 add x8,x8,x7	7:	add s0, s0, t2 # s0 = s..

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Registers

Name	Number	Value
sp	2	0x7ffffeffe
gp	3	0x100008000
tp	4	0x000000000
t0	5	0x000000000
t1	6	0xffffffff005
t2	7	0x000000000
s0	8	0x000000000
s1	9	0x000000000
a0	10	0x000000000
a1	11	0x000000000
a2	12	0x000000000
a3	13	0x000000000
a4	14	0x000000000
a5	15	0x000000000
a6	16	0x000000000
a7	17	0x000000000
s2	18	0x000000000
s3	19	0x000000000
s4	20	0x000000000
s5	21	0x000000000
s6	22	0x000000000
s7	23	0x000000000
s8	24	0x000000000
s9	25	0x000000000
s10	26	0x000000000
s11	27	0x000000000
t3	28	0x000000000
t4	29	0x000000000
t5	30	0x000000000
t6	31	0x000000000
pc		0x00400008

E:\Assembly\assignment.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed 1 Inst/sec

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00500313 addi x6,x0,5	3:	addi t1, zero, 5 # x = t1..
	0x00400004	0xffff00393 addi x7,x0,0xffffffff	4:	addi t2, zero, -1 # y = t2..
	0x00400008	0x00630433 add x8,x6,x6	6:	add s0, t1, t1 # s0 = t..
	0x004000c	0x00740433 add x8,x8,x7	7:	add s0, s0, t2 # s0 = s..

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Registers

Name	Number	Value
sp	2	0x7ffffeffe
gp	3	0x100008000
tp	4	0x000000000
t0	5	0x000000000
t1	6	0xffffffff005
t2	7	0xfffffffffff
s0	8	0x000000000
s1	9	0x000000000
a0	10	0x000000000
a1	11	0x000000000
a2	12	0x000000000
a3	13	0x000000000
a4	14	0x000000000
a5	15	0x000000000
a6	16	0x000000000
a7	17	0x000000000
s2	18	0x000000000
s3	19	0x000000000
s4	20	0x000000000
s5	21	0x000000000
s6	22	0x000000000
s7	23	0x000000000
s8	24	0x000000000
s9	25	0x000000000
s10	26	0x000000000
s11	27	0x000000000
t3	28	0x000000000
t4	29	0x000000000
t5	30	0x000000000
t6	31	0x000000000
pc		0x00400008

E:\Assembly\assignment.asm - RARS 1.6

Registers Floating Point Control and Status

Name	Number	Value
sp	2	0x7ffffeffe
gp	3	0x100000000
tp	4	0x000000000
t0	5	0x000000000
t1	6	0x000000005
t2	7	0xfffffffff
s0	8	0x000000004
s1	9	0x000000000
a0	10	0x000000000
a1	11	0x000000000
a2	12	0x000000000
a3	13	0x000000000
a4	14	0x000000000
a5	15	0x000000000
a6	16	0x000000000
a7	17	0x000000000
s2	18	0x000000000
s3	19	0x000000000
s4	20	0x000000000
s5	21	0x000000000
s6	22	0x000000000
s7	23	0x000000000
s8	24	0x000000000
s9	25	0x000000000
s10	26	0x000000000
s11	27	0x000000000
t3	28	0x000000000
t4	29	0x000000000
t5	30	0x000000000
t6	31	0x000000000
pc		0x0040000c

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00500313	addi x6,x0,5	3: addi t1, zero, 5 # x = t1..
	0x00400004	0xffff0039	addi x7,x0,0xffffffff	4: addi t2, zero, -1 # y = t2..
	0x00400008	0x00630433	add x8,x6,x6	6: add s0, t1, t1 # s0 = t..
	0x0040000c	0x00740433	add x8,x8,x7	7: add s0, s0, t2 # s0 = s..

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Run speed 1 inst/sec

Messages Run I/O

E:\Assembly\assignment.asm - RARS 1.6

Registers Floating Point Control and Status

Name	Number	Value
sp	2	0x7ffffeffe
gp	3	0x100000000
tp	4	0x000000000
t0	5	0x000000000
t1	6	0x000000005
t2	7	0xfffffffff
s0	8	0x000000004
s1	9	0x000000000
a0	10	0x000000000
a1	11	0x000000000
a2	12	0x000000000
a3	13	0x000000000
a4	14	0x000000000
a5	15	0x000000000
a6	16	0x000000000
a7	17	0x000000000
s2	18	0x000000000
s3	19	0x000000000
s4	20	0x000000000
s5	21	0x000000000
s6	22	0x000000000
s7	23	0x000000000
s8	24	0x000000000
s9	25	0x000000000
s10	26	0x000000000
s11	27	0x000000000
t3	28	0x000000000
t4	29	0x000000000
t5	30	0x000000000
t6	31	0x000000000
pc		0x0040000c

Text Segment

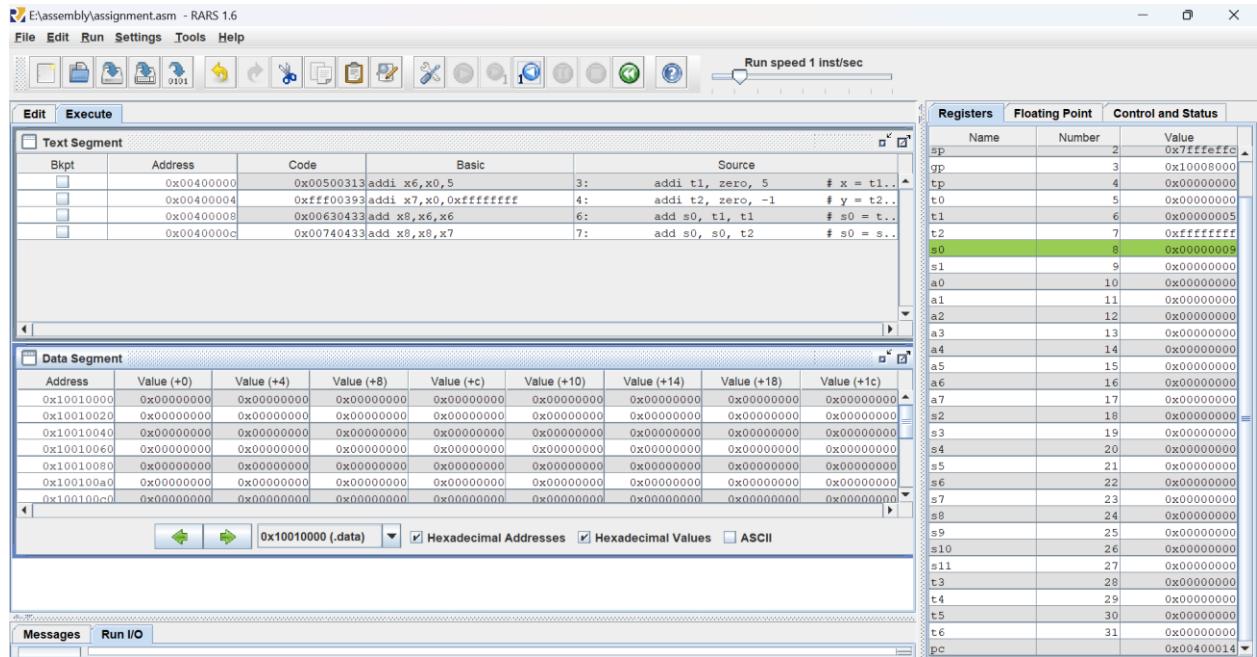
Bkpt	Address	Code	Basic	Source
	0x00400000	0x00500313	addi x6,x0,5	3: addi t1, zero, 5 # x = t1..
	0x00400004	0xffff0039	addi x7,x0,0xffffffff	4: addi t2, zero, -1 # y = t2..
	0x00400008	0x00630433	add x8,x6,x6	6: add s0, t1, t1 # s0 = t..
	0x0040000c	0x00740433	add x8,x8,x7	7: add s0, s0, t2 # s0 = s..

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Run speed 1 inst/sec

Messages Run I/O



The result of s0 is 9 correctly.

- The machine codes of addi (I-type) instructions: 0x00500313, 0xffff00393.
The machine codes of add (R-type) instructions: 0x00630433, 0x 00740433.
- For addi x6, x0, 5 instruction:
 - +) Opcode: 0010011 (7 bits)
 - +) rd (x6): 00110 (5 bits)
 - +) funct3: 000 (3 bits)
 - +) rs1 (x0): 00000 (5 bits)
 - +) imm[11:0] (5 in binary): 00000000101 (12 bits)
 - ⇒ In binary: 0000 0000 0101 0000 0000 0011 0001 0011
 - ⇒ Machine code in hexadecimal: 0x00500313

For addi x7, x0, 0xffffffff instruction:

- +) Opcode: 0010011 (7 bits)
- +) rd (x7): 00111 (5 bits)
- +) funct3: 000 (3 bits)
- +) rs1 (x0): 00000 (5 bits)
- + imm[11:0] (2's complement of 1 in binary): 111111111111 (12 bits)
 - ⇒ In binary: 1111 1111 1111 0000 0000 0011 1001 0011
 - ⇒ Machine code in hexadecimal: 0xffff00393

For add x8, x8, x6 instruction:

- +) Opcode: 0110011 (7 bits)
- +) rd (x8): 01000 (5 bits)
- +) funct3: 000 (3 bits)

+) rs1 (x6): 00110 (5 bits)

+) rs2 (x6): 00110 (5 bits)

+) funct7: 0000000 (7 bits)

⇒ In binary: 0000 0000 0110 0011 0000 0100 0011 0011

⇒ Machine code in hexadecimal: 0x00630433

For add x8, x8, x7 instruction:

+) Opcode: 0110011 (7 bits)

+) rd (x8): 01000 (5 bits)

+) funct3: 000 (3 bits)

+) rs1 (x8): 01000 (5 bits)

+) rs2 (x7): 00111 (5 bits)

+) funct7: 0000000 (7 bits)

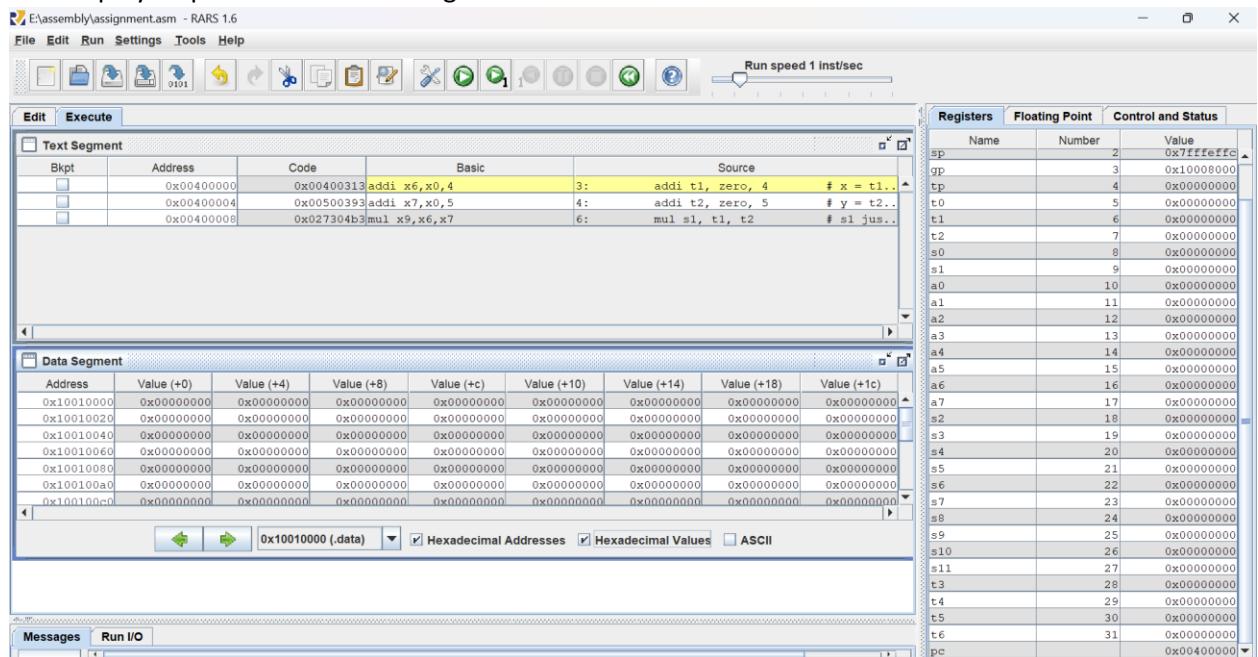
⇒ In binary: 0000 0000 0111 0100 0000 0100 0011 0011

⇒ Machine code in hexadecimal: 0x00740433

In conclusion, the result compiling manually is the same with machine code in Text Segment window.

Assignment 5: Multiplication

- Run step by step and monitor the Registers window:



E:\assembly\assignment.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed 1 inst/sec

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00400313 addi x6,x0,4	3:	addi t1, zero, 4 # x = t1..
	0x00400004	0x00500393 addi x7,x0,5	4:	addi t2, zero, 5 # y = t2..
	0x00400008	0x027304b3 mul s1, t1, t2	6:	mul s1, t1, t2 # s1 jus..

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Registers

Name	Number	Value
sp	2	0x7fffffecc
gp	3	0x100000000
tp	4	0x000000000
t0	5	0x000000000
t1	6	0x000000004
t2	7	0x000000000
s0	8	0x000000000
s1	9	0x000000000
a0	10	0x000000000
a1	11	0x000000000
a2	12	0x000000000
a3	13	0x000000000
a4	14	0x000000000
a5	15	0x000000000
a6	16	0x000000000
a7	17	0x000000000
s2	18	0x000000000
s3	19	0x000000000
s4	20	0x000000000
s5	21	0x000000000
s6	22	0x000000000
s7	23	0x000000000
s8	24	0x000000000
s9	25	0x000000000
s10	26	0x000000000
s11	27	0x000000000
t3	28	0x000000000
t4	29	0x000000000
t5	30	0x000000000
t6	31	0x000000000
pc		0x00400004

E:\assembly\assignment.asm - RARS 1.6

File Edit Run Settings Tools Help

Run speed 1 inst/sec

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00400313 addi x6,x0,4	3:	addi t1, zero, 4 # x = t1..
	0x00400004	0x00500393 addi x7,x0,5	4:	addi t2, zero, 5 # y = t2..
	0x00400008	0x027304b3 mul s1, t1, t2	6:	mul s1, t1, t2 # s1 jus..

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Registers

Name	Number	Value
sp	2	0x7fffffecc
gp	3	0x100000000
tp	4	0x000000000
t0	5	0x000000000
t1	6	0x000000004
t2	7	0x000000005
s0	8	0x000000000
s1	9	0x000000000
a0	10	0x000000000
a1	11	0x000000000
a2	12	0x000000000
a3	13	0x000000000
a4	14	0x000000000
a5	15	0x000000000
a6	16	0x000000000
a7	17	0x000000000
s2	18	0x000000000
s3	19	0x000000000
s4	20	0x000000000
s5	21	0x000000000
s6	22	0x000000000
s7	23	0x000000000
s8	24	0x000000000
s9	25	0x000000000
s10	26	0x000000000
s11	27	0x000000000
t3	28	0x000000000
t4	29	0x000000000
t5	30	0x000000000
t6	31	0x000000000
pc		0x00400008

Run 1 (Result 20):

```

E:\assembly\assignment.asm - RARS 1.6
File Edit Run Settings Tools Help
Run speed 1 inst/sec

Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 addi x6,x0,4 3: addi t1, zero, 4 # x = t1..
0x00400004 addi x7,x0,5 4: addi t2, zero, 5 # y = t2..
0x00400008 0x027304b3 mul s1, t1, t2 # s1 jus..

Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010020 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

Registers
Name Number Value
sp 2 0x7ffffeffc
gp 3 0x10008000
tp 4 0x00000000
t0 5 0x00000000
t1 6 0x00000004
t2 7 0x00000005
s0 8 0x00000000
s1 9 0x00000014
a0 10 0x00000000
a1 11 0x00000000
a2 12 0x00000000
a3 13 0x00000000
a4 14 0x00000000
a5 15 0x00000000
a6 16 0x00000000
a7 17 0x00000000
s2 18 0x00000000
s3 19 0x00000000
s4 20 0x00000000
s5 21 0x00000000
s6 22 0x00000000
s7 23 0x00000000
s8 24 0x00000000
s9 25 0x00000000
s10 26 0x00000000
s11 27 0x00000000
t3 28 0x00000000
t4 29 0x00000000
t5 30 0x00000000
t6 31 0x00000000
pc 32 0x0040000c

Messages Run I/O

```

Run 2 (Result 21):

```

E:\assembly\assignment.asm - RARS 1.6
File Edit Run Settings Tools Help
Run speed 1 inst/sec

Edit Execute
Text Segment
Bkpt Address Code Basic Source
0x00400000 addi x6,x0,4 3: addi t1, zero, 4 # x = t1..
0x00400004 addi x7,x0,5 4: addi t2, zero, 5 # y = t2..
0x00400008 0x027304b3 mul s1, t1, t2 # s1 jus..

Data Segment
Address Value (+0) Value (+4) Value (+8) Value (+c) Value (+10) Value (+14) Value (+18) Value (+1c)
0x10010000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010020 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010040 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010060 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x10010080 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100a0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
0x100100c0 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000

Registers
Name Number Value
sp 2 0x7ffffeffc
gp 3 0x10008000
tp 4 0x00000000
t0 5 0x00000000
t1 6 0x00000004
t2 7 0x00000005
s0 8 0x00000000
s1 9 0x00000014
a0 10 0x00000000
a1 11 0x00000000
a2 12 0x00000000
a3 13 0x00000000
a4 14 0x00000000
a5 15 0x00000000
a6 16 0x00000000
a7 17 0x00000000
s2 18 0x00000000
s3 19 0x00000000
s4 20 0x00000000
s5 21 0x00000000
s6 22 0x00000000
s7 23 0x00000000
s8 24 0x00000000
s9 25 0x00000000
s10 26 0x00000000
s11 27 0x00000000
t3 28 0x00000000
t4 29 0x00000000
t5 30 0x00000000
t6 31 0x00000000
pc 32 0x00400010

Messages Run I/O

```

- The result of multiplication instructions is 20 in decimal correctly.
- Try division instructions:

The screenshot shows the RARS 1.6 assembly editor interface. The code window displays the following assembly code:

```

1 # Laboratory Exercise 2, Assignment 5
2 .text
3     addi t1, zero, 10      # x = t1 = ?
4     addi t2, zero, 5       # y = t2 = ?
5
6     div s1, t1, t2
7

```

The Registers window shows the following initial values:

Name	Number	Value
sp	2	0x7ffffefffa
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400000

The screenshot shows the RARS 1.6 assembly editor interface. The code window displays the same assembly code as the previous screenshot.

The Labels window shows the following entries:

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00a00313	addi x6,x0,10	3: addi t1, zero, 10 # x = t1 = ?
	0x00400004	0x00500393	addi x7,x0,5	4: addi t2, zero, 5 # y = t2 = ?
	0x00400008	0x027344b3	div x9,x6,x7	6: div s1, t1, t2

The Data Segment window shows the following memory dump:

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

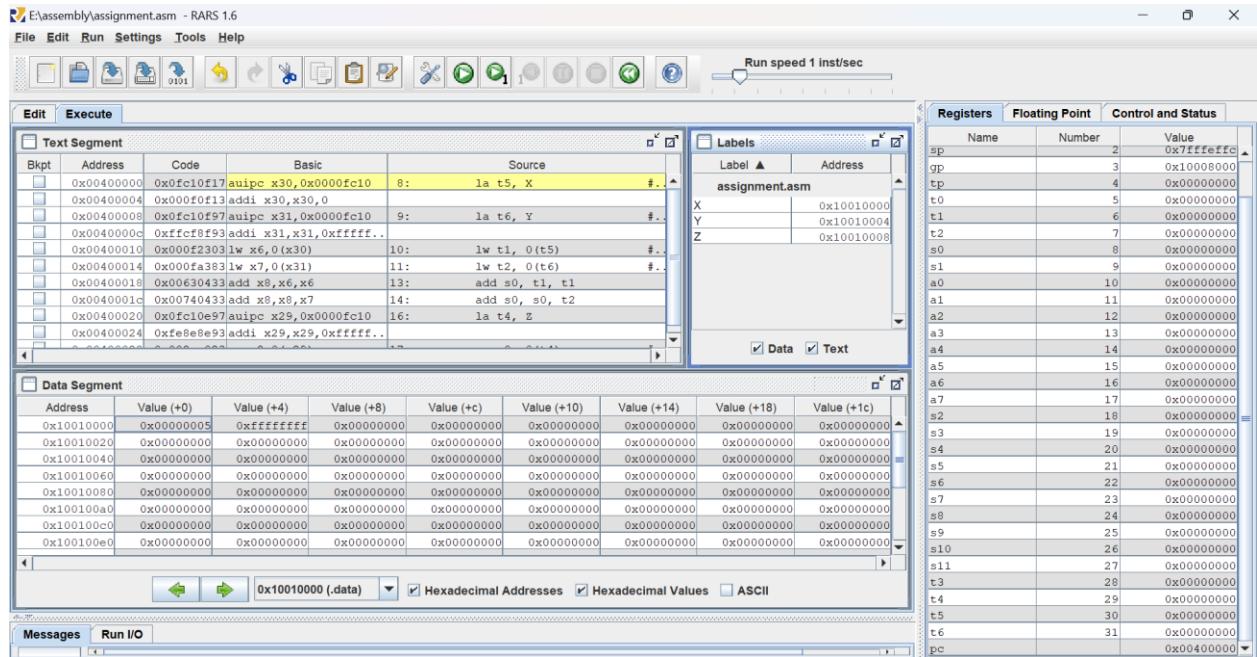
The Registers window shows the following initial values:

Name	Number	Value
sp	2	0x7ffffefffa
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000005
s0	8	0x00000000
s1	9	0x00000002
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400010

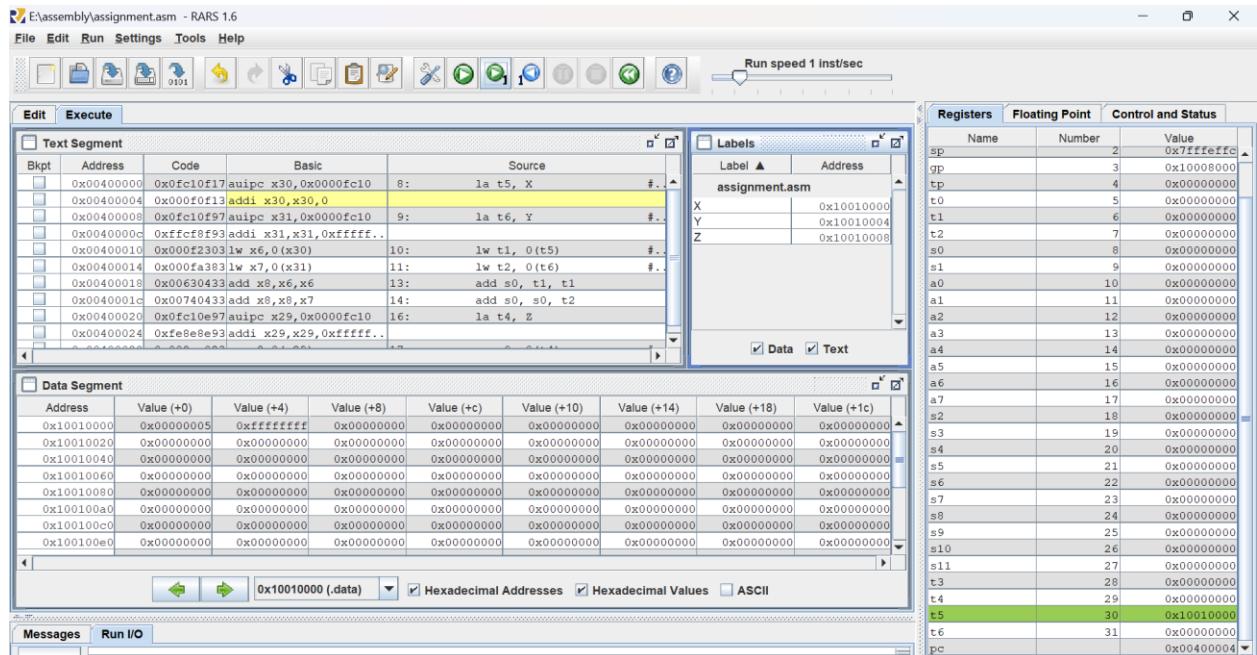
Assignment 6: Declare and access variables

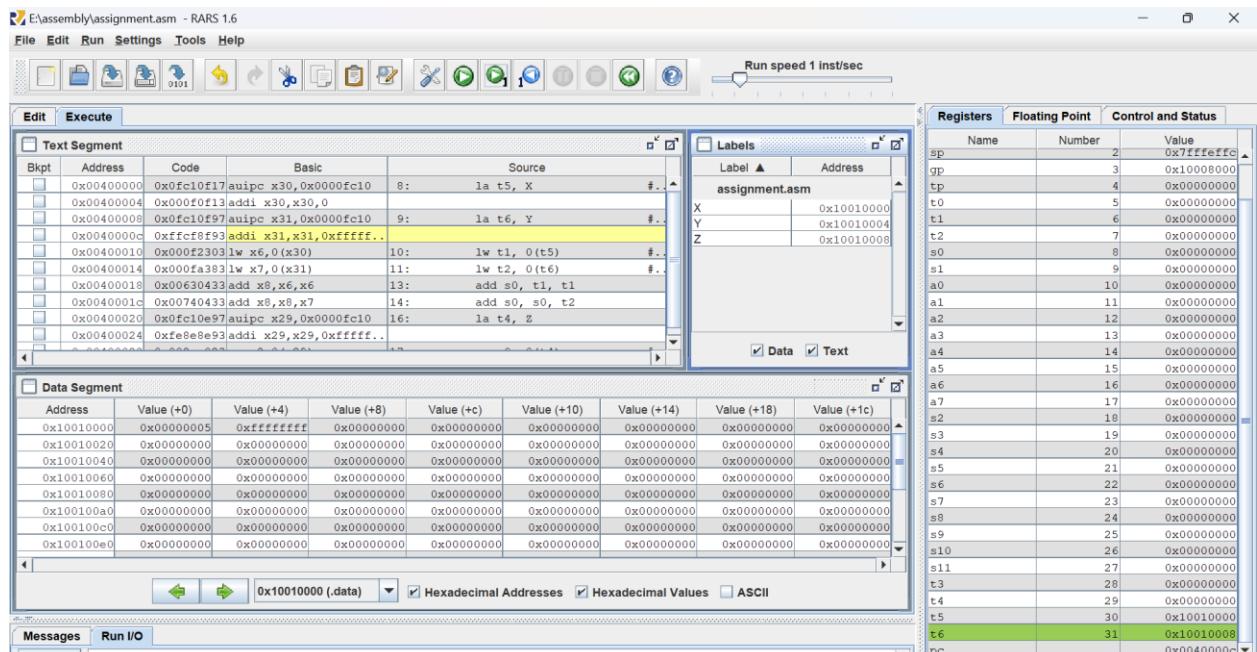
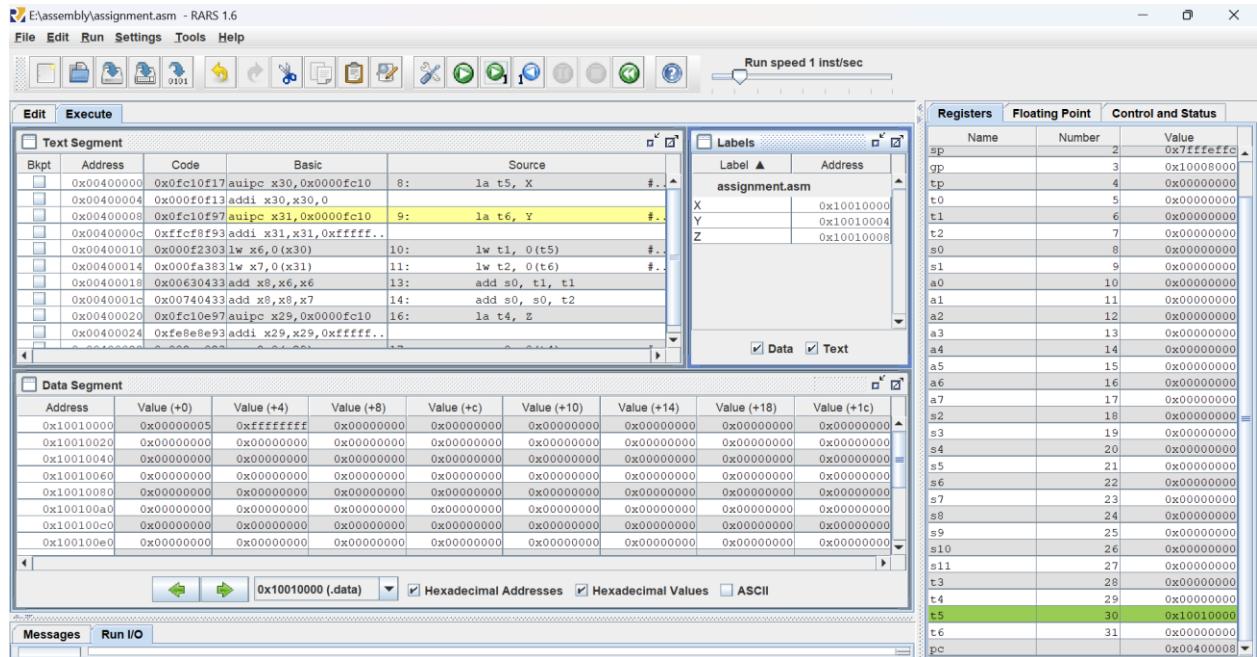
- The Labels window:

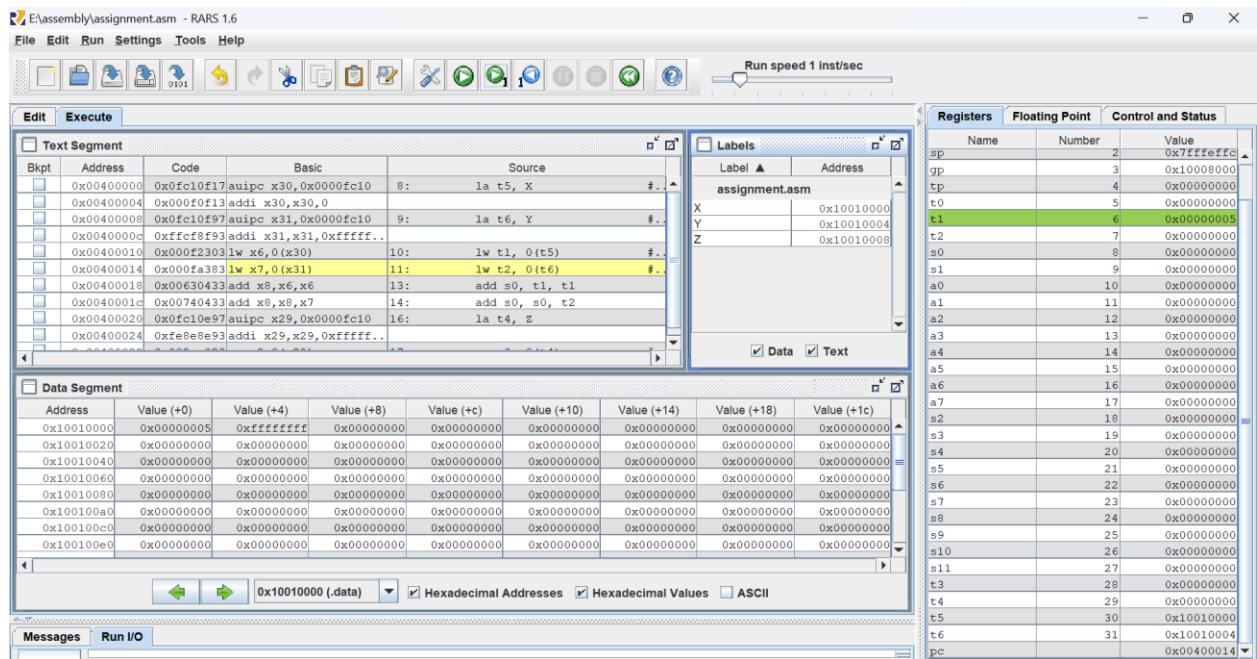
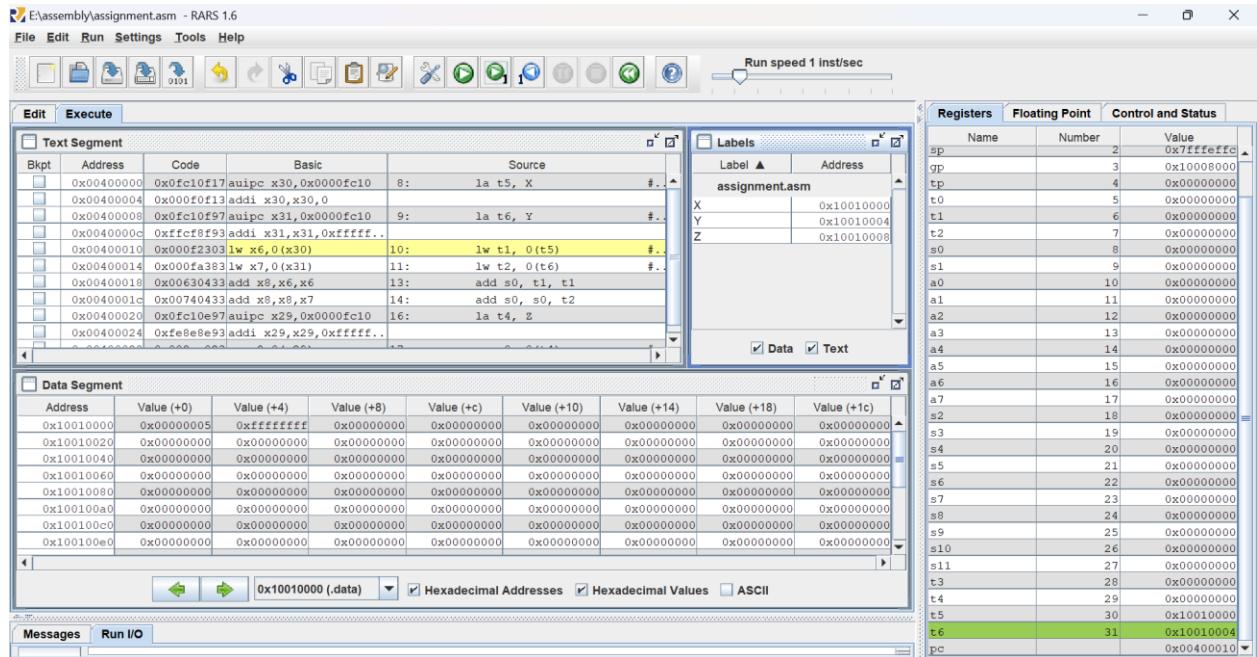
The values in Data Segment are the same with initial values respectively.

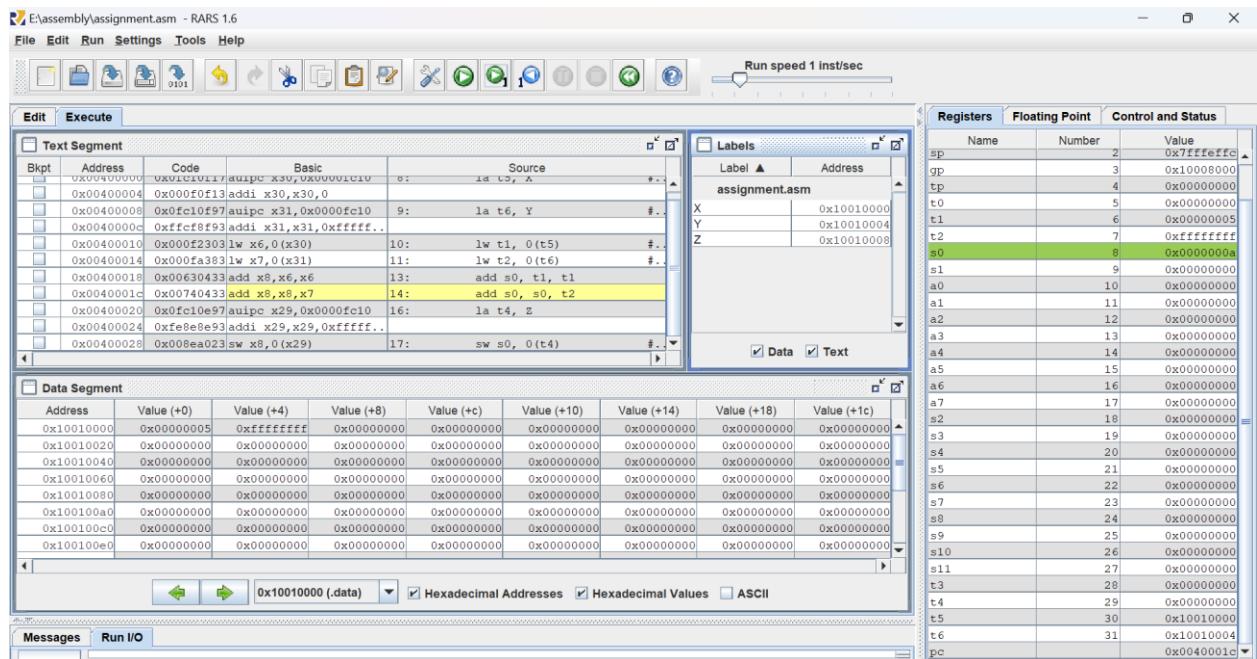
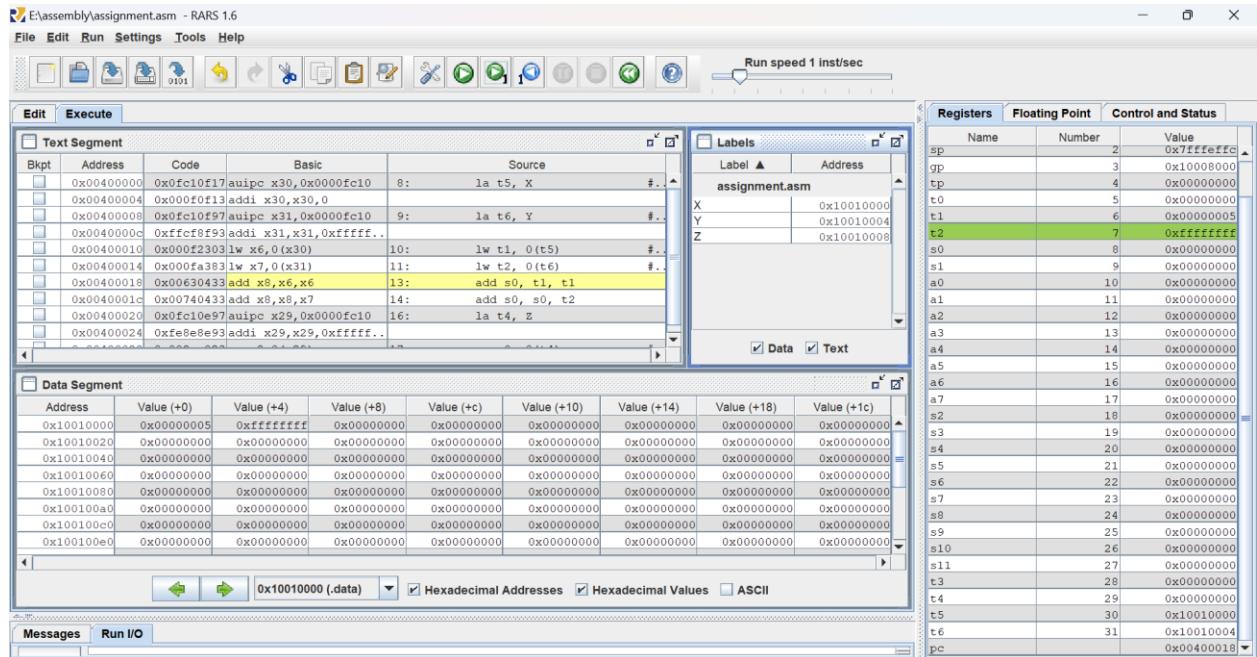


- La instruction includes components: targets register, address, offset, base register. The instruction computes the effective address by adding the offset to the value in the base register. The computed address is then stored in the target register.
- Run step by step and monitor the Registers window:



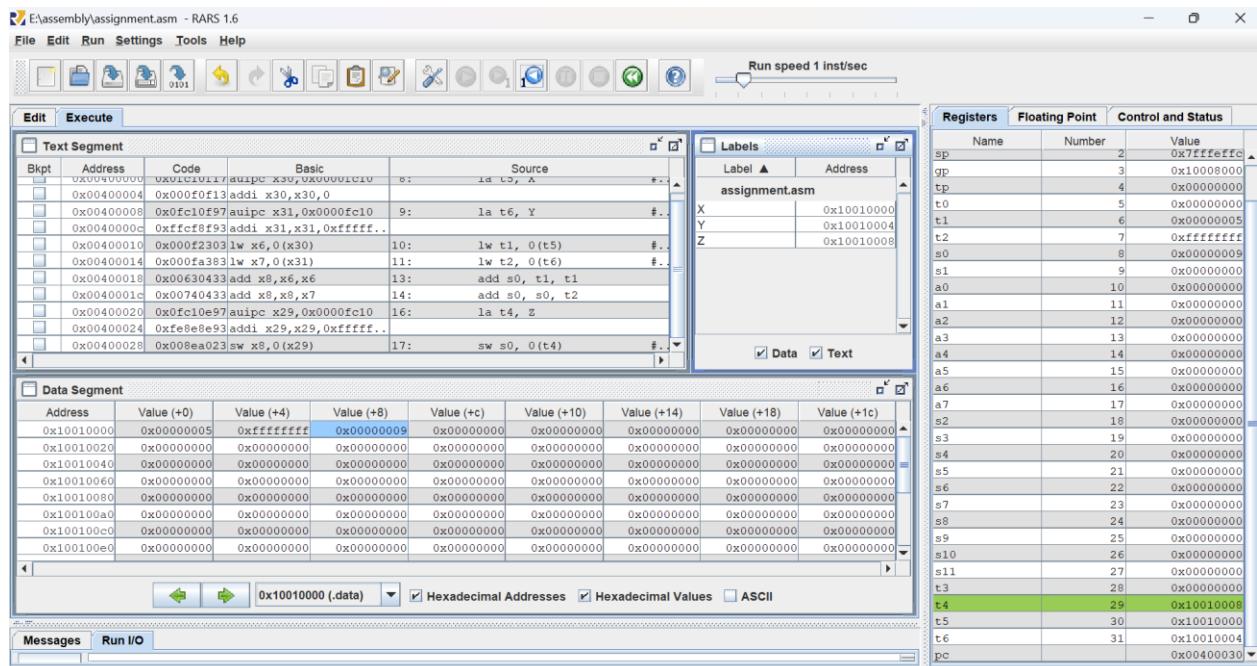
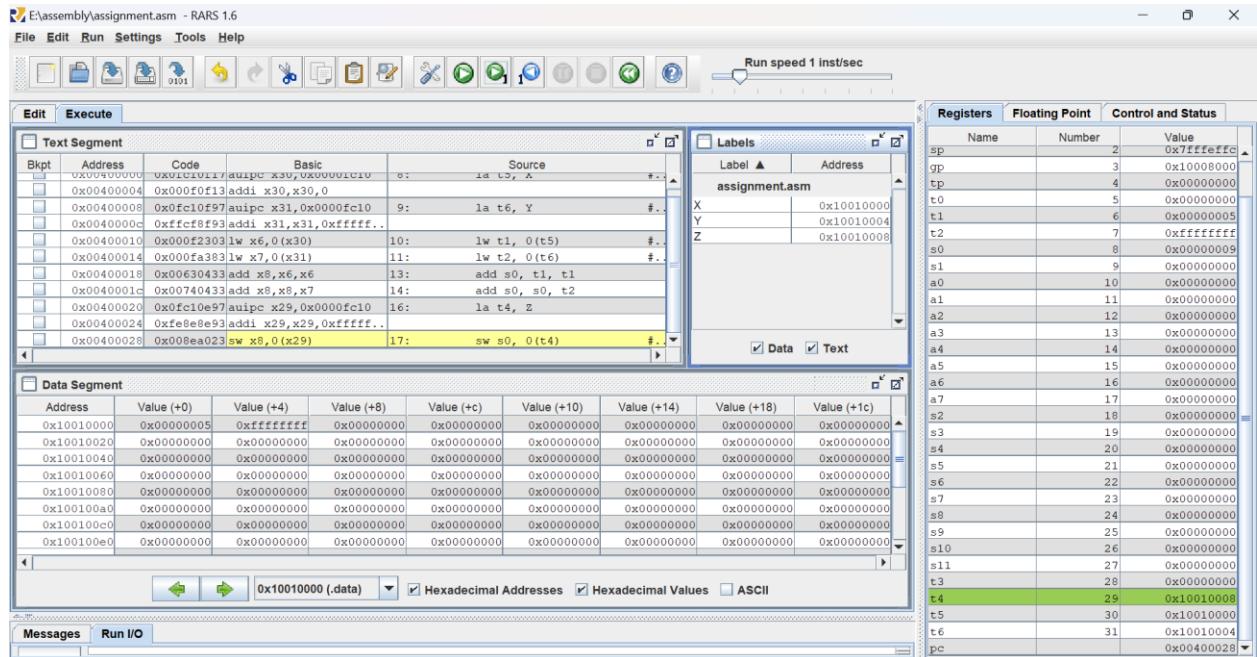






Name	Number	Value
sp	2	0x7ffffefffa
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000005
t2	7	0xffffffff
s0	8	0x00000008
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x10010000
t6	31	0x10010004
pc		0x00400020

Name	Number	Value
sp	2	0x7ffffefffa
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000005
t2	7	0xffffffff
s0	8	0x00000008
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x10010020
t5	30	0x10010000
t6	31	0x10010004
pc		0x00400024



Assignment 7: Declare variables or instructions at specified address

E:\assembly\assignment.asm - RARS 1.6

Registers

Name	Number	Value
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400004

Registers

Name	Number	Value
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400004

E:\assembly\assignment.asm - RARS 1.6

Registers

Name	Number	Value
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400004

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00408000	0x00200093	addi xl,x0,2	7: addi xl, zero, 2

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Registers

Name	Number	Value
sp	2	0x7ffffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400004